Developing a Robust WWW Interface to Scholarly Databases: HTML (PHP+mSQL+OpenWeb) = Increased Functionality

Faith McGrath R.R.T., M.L.S. and Kathryn Latimer - Yale University School of Medicine

Public access computers in the medical library and hospital have traditionally used proprietary software with locally written scripts to provide a menu driven interface to high use medical center resources and to log usage statistics for these resources. Ubiquitous access to Ethernet, Novell, PPP and ARA has drastically increased access to scholarly resources once only available within the library or from clinical workstations on patient wards. Proprietary software is expensive and it is untenable to support locally loaded software upgrades and often changing scripts on hundreds of PCs, not to mention the multitudes of home, office and dorm computers. Multiple operating systems add another layer of support.

The WWW client/server paradigm which is OS independent, non-proprietary, centrally maintained, easily and freely available to end users, would seem to be a better interface solution for both public workstations and remote users. The beauty of this interface is its seamless access to resources from a familiar GUI hypertext interface which is available to anyone with Internet access and a WWW browser, but on closer scrutiny there is loss of some functionality. Unfortunately, important usage statistics for individual databases/resources (not simply page hits) are no longer available for analysis. Resources which require launching of executable programs and/or networked CD ROMs are no longer accessible. There are also security issues with the client software on public access machines.

To increase the functionality of our basic HTML WWW site (http://www.med.yale.edu/medmenu/) we are using PHP/FI (Personal Home Page/Form Interpreter) and mSQL (mini-Simple Query Language) which provides functionality similar to proprietary software with the easy of client/server maintenance. Its server-side HTML-embedded language allows you to write simple scripts directly in your HTML files much like JavaScript does, except, unlike JavaScript it is not browser-dependant. JavaScript is a client-side embedded language while PHP/FI is a server-side language. PHP/FI comes with full source code. MSQL is a simple SQL engine for UNIX systems. Both are free to non-commercial sites. PHP/FI provides support for embedded SQL right from a HTML file. This provides a simple mechanism to build a database-enabled WWW page system with the combination of mSQL and PHP/FI. The intent of the database is to have a single location to add and/or update the resource information and still allow it to be viewed in a variety of formats. Maintenance is form based and requires no HTML experience. The menuing system is based on a 'top' level menu that contains heavily used, or popular items and the 'list' level menu that displays a comprehensive list of high use resources. Running PHP/FI as an Apache (public domain HTTP server for Unix) module is the most efficient way of using the package. PHP is compiled into a module and registered with the Apache server. This method saves on CPU cycles and memory, as each call to a PHP/FI script does not launch its own external process. Multiple PHP/FI scripts can be executed simultaneously. PHP compilation provides for logging of statistics to monitor resource usage. PHP (as well as HTML) redirects the page first and then logs the data to minimize the loss of speed for the user. In the past, locally loaded scripts provided good statistics for library and hospital public usage, but since the vast majority of our users are remote (labs, offices, dorms and homes etc.,) PHP allows us to view statistics from a much larger pool of users. It also allows us to centrally design multiple versions (i.e., http://www.med.yale.edu/medmenu/top.phtml?version=r for the PCs in the library) of the site for different audiences, such as a Macintosh education software users, a library PC users and 'the world'.

We now have a more robust WWW site and there is a universal interface for public and remote computers, but there are some applications which are still only accessible from public workstations. Several methods for launching CD-ROMs have been discussed in library presentations, but we selected OpenWEB netDeploy because it supports both Macintosh and Windows. Benefits of this system allow server mounted software to be accessible via the WWW without changing the client setup and a simple method of building the file of MIME type (application/openweb) to be located on the WWW server. Applications and batch files can be launched without loading Windows File Manager (security risk) and administration is centralized. Surveys written in Visual Basic can be launched via OpenWeb as regular server mounted applications to obtain user input from the public computers. NetDeploy is freely available to non-commercial institutions. Security is a concern for publicly accessible workstations. We chose a customized kiosk mode which uses a resource editor to limit netscape.exe. This method requires changes when the browser is updated and is WINDOWS specific. We used the ResEdit (shareware) program to provide the same functionality on the Macintosh.

Further documentation: PHP/FI Version 2.0 - http://info.med.yale.edu/docs/php/ and MedMenu PHTML/MSQL ReadMe - http://www.med.yale.edu/medmenu/medmenu.readme.html.

1091—8280/97/$5.00